IN THE SPECIFICATION:

Please add the following paragraph immediately following paragraph 19:

[0019.1] FIG. 9A illustrates exemplary pseudo-code for a procedure call and illustrates a procedure declaration of the procedure called by the procedure call.

Please replace paragraphs 0020, 0021, 0037, 0042, and 0048 with the following amended paragraphs:

[0020] FIG. ~~9A~~ 9B depicts an exemplary process to implement a procedure call using a memory-based style of call linkage; and

[0021] FIG. ~~9B~~ 9C depicts an exemplary process to implement a procedure call using a register-based style of call linkage.

[0037] In general, the routines executed to implement ~~the~~ embodiments of the invention, whether implemented as part of an operating system or a specific application, component, program, object, module or sequence of instructions, are in the compiler program 110~~, or the compiler 110~~ (or "compiler" for short). The compiler 110 typically comprises one or more instructions that are resident at various times in various memory and storage devices in the computer system 100. When read and executed by one or more processors 102 in the computer system 100, the compiler 110 causes that computer system 100 to perform the steps necessary to execute steps or elements embodying the various aspects of the invention. Moreover, while the invention has and hereinafter will be described in the context of fully functioning computers and computer systems, those skilled in the art will appreciate that the various embodiments of the invention are capable of being distributed as a program product in a variety of forms, and that the invention applies equally regardless of the particular type of signal bearing or computer readable media used to actually carry out the distribution. Examples of signal bearing or computer readable media include, but are not limited to, recordable type media such as volatile and nonvolatile memory devices, floppy and other removable disks, hard disk drives, optical disks (e.g., CD-ROM, DVD, and the like), among others.

[0042] FIG. 3 depicts one representation of the source code 114. The source code 114 is generally written in a high-level programming language. The source code 114 is only exemplary in nature and does not limit the scope of the present invention. The source code 114 comprises three modules, module A 202_1, module B 202_2 and module C 202_3. Each module 202_1, 202_2 and 202_3 comprises procedures, i.e., procedure definitions, and procedure calls. For example, module A 202_1 comprises procedures calcsum and calcprd, module B 202_2 comprises procedures sum and prd, and module C 202_3 comprises procedure main. Additionally, module A 202_1 comprises four procedure calls including two procedure calls to the callee procedure sum and two procedure calls to the callee procedure prd. Module C 202_3 comprises two procedure calls to the callee procedures calcsum and ~~calcprod~~ caclprd. These procedure calls may comprise intermodule calls. Namely, a procedure call in one module may call a procedure in a different module. For example, a procedure call of procedure calcsum is in module C 202_3 which calls the procedure calcsum in module A 202_1.

[0048] At step 508, the method 500 uses the data structures 117 to optimize the call linkages. More specifically, step 508 optimizes the call linkages by determining an appropriate call linkage for each procedure call, e.g., caller function and called procedure. [[.]] The call linkages are optimized to minimize the overall run time of the object code 116 generated from the source code 114. Step 508 is further described with respect to FIG. 8. The method 500 proceeds generate the object code 116 from the source code 114 at step 510 and ends at step 512.